AMENDMENTS TO THE CLAIMS

Listing of the Claims:

This listing will replace all prior versions, and listings, of claims in the application.

1. (Cancelled)

Claim 2. (Currently Amended) ~~A method as claimed in Claim 1~~

A method for use by a host microprocessor which translates sequences of instructions from a target instruction set for a target processor to sequences of instructions for the host microprocessor comprising the steps of:

beginning execution of a first sequence of target instructions by committing state of the target processor and storing memory stores generated by previously-executed sequences of instructions at a point in the execution of instructions at which state of the target processor is known,

beginning execution of a speculative sequence of host instructions following a branch from the first sequence of target instructions by immediately committing state and storing memory stores,

attempting to execute the speculative sequence of host instructions until another point in the execution of target instructions at which state of the target processor is known,

rolling back to last committed state of the target processor and discarding memory stores generated by the speculative sequence of host instructions if execution fails,

beginning execution of a next sequence of target instructions if execution succeeds, and

~~including an additional step of~~ releasing a lock for any sequence of host instructions running in a locked condition immediately after committing state of the target processor and storing memory stores generated by previously-executed translation sequences.

Claims 3-5. (Cancelled)

6. (Currently Amended) ~~A method as claimed in Claim 5~~

A method for use by a host microprocessor which translates sequences of target instructions from a target instruction set for a target processor to sequences of host instructions for the host microprocessor comprising:

committing state of the target processor and storing memory stores generated by previously executed sequences of host instructions at a point in execution of host instructions at which state of the target processor is known; and

releasing a lock for any sequence of host instructions running in a locked condition immediately after committing state of the target processor and storing memory stores generated by previously-executed translation sequences.

7. (Cancelled)

8. (Previously Presented) A method of executing instructions in a host microprocessor which translates sequences of instructions from a target instruction set for a target processor to sequences of instructions for the host microprocessor, said method comprising:

if a sequence of instructions includes a locking operation, placing a commit operation at the beginning of the sequence of instructions including the locking operation; and

if a sequence of instructions does not include a locking operation, placing a commit operation at the end of the sequence of instructions not including the locking operation.

9. (New) The method of Claim 2, further comprising storing an indication of lock state when state of said target processor is committed.

10. (New) The method of Claim 6, further comprising storing an indication of lock state when state of said target processor is committed.

11. (New) The method of Claim 8, further comprising storing an indication of lock state when state of said target processor is committed.